iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** V **Month of publication:** May 2020

DOI: http://doi.org/10.22214/ijraset.2020.5494

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue V May 2020- Available at www.ijraset.com

Neural Machine Translation from English to Hindi

Aditya Mittal[1], Mayank Wadhwa[2], Mr. Amit Chug[3]
[1, 2, 3]*Manav Rachna Institute of a Research and Studies*

Abstract: Language translation is the task in which machine is facing lagging behind the cognitive powers of human beings. Statistical Machine Translation is solving the problem of machine translation but it requires huge data sets and performs well on similar grammar structured language pairs. Now, Neural Machine Translation has emerged as an alternate way for doing same. In this paper, we explore Neural Machine Translation System from English to Hindi. We compared our results with machine translation techniques. We observed in this work that NMT requires very less amount of data size for training.
Keywords: Neural machine translation, long and short term memory, algorithm, encoder decoder model.

I. INTRODUCTION

Machine translation is one of the tasks which is taken by the computer scientists and in this field research is going past 50 years. This is a remarkable progress that our engineers of computer have worked together to achieve this goal current status of machine translation. Most of the machine translation are based on the statistical machine translation. In this system the general units of processes of translation are sentences and phrases.in these pairs one face belongs to source language and other is the target language but the probability is low of pairing and predicting the correct pair.

In recent years, Google has shifted it translation towards Neural machine translation proposed a sequence to sequence learning mechanism using LSTM means long and short term memory models. This neural network based machine translation system has eight layer of encoder and decoder which have same eight layer .A bidirectional recurrent neural network known as encoder.

Normally this translation requires a lot of computing power it means that normally a great technique if we have computing powers or enough time.

A. End to end Training

All parameters are simultaneously optimized to minimize a loss function on the network output.

B. Distributed Representations Share Strength

C. Better Exploitation Of Word And Phrase Similarities

1) Better exploitation of context
2) Neural machine translation uses a very bigger context both source and partial target text- to translate.
3) More fluent text generation
4) Deep learning text generation is much higher quality.

II. NEURAL MACHINE TRANSLATION

Neural machine translation uses bidirectional recurrent neural network. It is also called an encoder which processes a source sentence into vectors for a second recurrent neural network, which is called the decoder. And this decoder is used to predict the words in the target language. This process while we are differing from phrase based models method and hence proved to be very comparable in speed and its accuracy.

This machine translation approaches very large artificial neural network predicting the sequence of words in the form of complete sentences. Statistical machine translation consumes more time and our space means memory. Neural machine translation trains it to parts end to end to increase the performance.

1) Neural machine translation or NMT for shorts, is the use of neural network models to learn a statistical model for machine translation.
2) The key benefit to the approach is that a single system can be trained directly on source and target text.
3) This model consists of two main components-

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue V May 2020- Available at www.ijraset.com

A. Translation Model

1) Trained from parallel corpora
2) Maps source to target language.
3) Associated with adequacy

B. Language Model

1) Trained from monolingual data
2) Distinguishes good target usage
3) Associated with fluency

III. ENCODER –DECODER MODEL

An neural network encoder encodes and read the source sentence in to a fixed amount of length vector. This decoder gives the output a translation from encoded vector. The full encoder and decoder system is connected trained and to maximize the probability of right translation from the given source.
Source sentence= S=(s1,s2,s3---sn)
Target sentence= T=(t1,t2,t3---tn)
All sentences of varying length are encoded into fixed sized vector.
Uses fraction of the memory needed by traditional SMT models.
Performance of this model decreases as the length of a source sentence increase.
Uses RMN for both encoding and decoding.
Encoder maps the variable length sentence into a fixed length vector.
Decoder translates the vector representation back into a variable length target sequence.
Two networks are trained jointly to maximize the conditional probability of the target sentence, given the source sentence.
This model learns a continuous space representation of a phrase that preserves both the semantic and syntactic structure of the phrase.

A. Natural Language Process

Natural language processing is a type of artificial intelligence in which computer can understand, analyze and create meaning like human language in a better way. Developers do much operations using NLP, such as entity recognition, automatic summary, relationship extraction, translation, topic segmentation and speech recognition.

Uses of NLP algorithm are:

1) Summarize blocks of text.
2) Create a chat box.
3) Automatically generate keyword tags.
4) Identify the type of entity extracted.
5) Identify the sentiment of a string of text.
6) Reduce words to their root.

Natural language processes is said that the manipulations of natural language is doing automatic.
It works like human beings communicate each other i.e. speech, text. We speak to each other, as we write more. But it is easier to learn to speak even to write.
We communicate each other by text and voice. We have methods of natural language to understand for other types of data.

a) How human communicate each other
b) Computer should replicate the same thing.
c) Applications
d) Speech recognition
e) Machine translation
f) Chat boots etc.

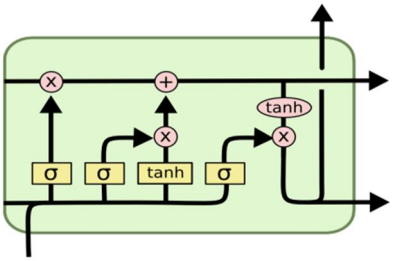
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue V May 2020- Available at www.ijraset.com

IV. LONG SHORT TERM MEMORY

The key recurrent components and LSTM memory cells let say there are some ten or units in the cell that may be two units can carry their long term information. And the rest of units can carry short term memory information which is why it is called long short term memory. The cells are controlled by gates. The forget gate is deciding which you need to keep short term memory and which one to keep long term memory and the sigma gating is multiplied to the previous cells value. The gating is learnt in terms of linear combination of waited inputs and hidden units. Think of the input and previous hidden units as dominant controllers what is deciding all kinds of gating. The same gate is again decided by the same controllers. The difference is only the terms of weight matrix and the placement. It controls the influx of information which is why it is called input gate. The influx is based on the 10x version of weighted combination of input and previous hidden units. The last gate is the output gate where the weight combination of controllers decided what is the next hidden units . Now, one popular variants of this model tries to try the not this model input and forgot gate. It means you only modify the short term cell units which are need to be forgotten. The input and forgotten gates are therefore complementary to each other and the gates. It not only compares information and for but even get to the separate by memory cell as you can observe that 10x version of the previous unit as impact.

The couples impact an the forgotten gates are is what they called as research gate and it decides the feet of each unit of cell. It can decide whatever the hidden unit or will they carry the long information.

LSTM have the structure like chain but the structure of this module repeating is different from RNM. LTSM module shown in fig:



A. History of Machine Translation

1) LATE 1940's : Early rule based systems
2) 1966: ALPAC report cuts research funding.
3) Early 1970's: First commercial systems.
4) Late 1980's: IBM develop first statistical models, inspired by speech research.
5) Early 2000's: Explosion in MT research.
6) 2006: First version of Moses.

B. Limitations Of Neural Machine Translation

1) Clarity of the text (ambiguity issues)
2) Human judgement and the training (legal)
3) Privacy of the data management
4) Creativity aspect

C. Applications

Over the world the most popular translation machine in the whole world is Google translate. This system uses Google neural machine translation to increases accuracy and most important its fluency. This system does not applies a large data set for algorithm to its training. It work with end to end design and it allows the system to learn over time and can create do better, more neural translations.

1) Flexible
2) Retained layout
3) Quick turnaround
4) High security

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue V May 2020- Available at www.ijraset.com

V. ALGORITHM

This algorithm can translate in the real time and the probability of sequences of word. It enables the translation engine to learn to translate of neural network structured connections like human brain. Neural machine translation do best quality translate which produced by these systems containing 17 percent of fewer lexical errors and 50 percent of fewer word and 19 percent of grammatical errors.

It has even learned to correct match the gender and different languages. Here, we have an example of a phrase translated from English to Hindi by neural machine translation that engine has been trained in this field.

NMT employs use of the words of vector representations and the internal state. It means that the words are translated into a vector defined by a separate magnitude and unique direction. If we compared to phrase based model this work is more simpler than if we compared with separate component like the language model and model named language model and mainly uses a single sequence model which produces one word at a time.

A. Section I: The Corpus — Pre-processing

1) Dealing with sentence length outliers (some sentences are clubbed together and end in the entire sequence length being 2000, which is grammatically illogical! It is always better to urge obviate those, specially during this case, there have been 4 such sentences.)

2) Basic text cleaning bad data in files (quotes from other languages), copyright statements and e-mail addresses (need no translation), punctuation, digits, converting to small letter etc..

3) Append “START_” & “_END” to the target sentences to English. We will see the reason for doing this in subsequent sections.

Pre-processing consumes most of your time, esp. from a knowledge Science perspective, you'd need a detailed study of the info. Such analyses give better insights and help make better decisions. For instance, I found that the typical sentence lengths in both languages were 14 and 15, which made me consider sentences up to length 30. There can be many similar decisions that you need to take before you go on to design your model.

B. Section II: Preparing Training Data

We still have the data in text format. We need to form it machine-ready for training our model. So, before model design, we'll perform Tokenizing and Indexing. For tokenization, we will find all the unique words in both languages. This will determine the dimensionality of the index arrays. Now we create 3 numpy arrays and encoder for input, decoder for input and decoder for target. We will use this for indexing each word. In the above step, the size 30 and 32 are due to the utmost sentence lengths we've decided. it's 30 for encoder (Hindi) and 32 for decoder (English). Decoder limit is 32 because “START_” and “_END” are appended to the start and therefore the end of the target sentences (in this case, English), in order that the decoder features a stopping condition which is either an “_END” being encountered, or maximum word limit is reached. Also, the “START_” is employed because decoder output are going to be one time-step ahead.

C. Section III: Word Embeddings

This model on text data, we'd like to possess a featurized representation for every word , also referred to as word embedding. the thought is to find out a group of features and their values, in order that we've dense vector representations for words. The most common example to elucidate word embeddings is that the ‘Man-Woman-King-Queen’ example. If ‘Man’ is at the position 5545 during a vocabulary of size 10,000, the one-hot vector for ‘Man’ may be a 10,000-dimensional vector of 0s, with just one entry as ‘1’ i.e. at the position 5545, denoted as O_{5545}. Now, if we define these words by some features, say 50, we will define a matrix of the values of all 50 features, for every word within the corpus. an off-the-cuff representation of the ’Embedding matrix. This matrix is initialized randomly. ‘Gender’, ‘Royal’ and ‘Kind’ are 3 out of fifty features. So, each word from this matrix are going to be a 50-dimensional vector. for instance , if the male gender is taken into account as -1 and feminine as 1 within the embedding matrix, the word ‘Man’ features a value -1 for the ‘Gender’ feature and ‘King’ has -0.99 for an equivalent feature. So, when these word vectors are going to be placed during a 50-dimensional vector space, there are high probabilities of a greater cosine similarity in between vectors, if all other features are similar too. Now, to seek out the word embedding of ‘Man’, the embedding matrix is multiplied with the one-hot encoded vector(O_{5545}) for ‘Man’.

$$\mathbf{E}_{50\times 10K} = \begin{array}{c} \begin{matrix} \text{Man} & \text{Woman} & \text{King} & \text{Queen} \end{matrix} \\ \begin{bmatrix} -1 & 1 & -0.99 & 0.98 \\ 0.02 & 0.01 & 0.94 & 0.97 \\ \vdots & \vdots & \vdots & \vdots \\ 0.78 & 0.84 & 0.91 & 0.92 \end{bmatrix} \end{array} \begin{matrix} \\ \text{Gender} \\ \text{Royal} \\ \\ \text{Kind} \end{matrix}$$

$$\begin{array}{c} \begin{matrix} \text{Man} & \text{Woman} & \text{(All words)} & \text{King} & \text{Queen} \end{matrix} \\ \begin{bmatrix} -1 & 1 & \dots & -0.99 & 0.98 \\ 0.02 & 0.01 & \dots & 0.94 & 0.97 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.78 & 0.84 & \dots & 0.91 & 0.92 \end{bmatrix}_{50\times 10K} \end{array} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \boxed{1} \\ \vdots \\ 0 \end{bmatrix}_{10K\times 1} = \begin{bmatrix} -1 \\ 0.02 \\ \vdots \\ 0.78 \end{bmatrix}_{50\times 1}$$

D. Section IV: Sequence to Sequence Learning

Give the vectors for source sequences in Hindi, to the encoder network, each word at a time.

Then encode the input sentences into a fixed dimension vectors then, we get the hidden and cell states from the encoder LSTM, and give it to the decoder LSTM.

These states considered to be initial states by decoder and has the embedding vectors for target words in English.

Decode the target of output the translated sentence, eachword at a time. during this step, the output of the decoder is shipped to a softmax layer over the whole target vocabulary.

1) *Encoder LSTM:* The important thing to notice here is that the input sequences are going to be of various lengths. So, to take care of a continuing length of inputs, the utmost length of sentences is calculated and therefore the input matrix dimension is chosen accordingly. Encoder LSTM processes the input sequence and returns the interior states. At this stage, the input sequence is mapped to a hard and fast dimension state vector, which is further fed to the decoder LSTM as its initial state.
2) *Decoder LSTM:* The decoder LSTM takes as input the state vector mapped by the encoder, and it's then trained to output the interpretation , one word at a time. This LSTM predicts subsequent words of the target sequence, given the previously translated words from the sequence. It basically generates the target sequence, offset by one time-step in future, using the state vectors from encoder because the initial state.

E. Section V: Prediction — Beam Search

P(Y | X) where (Y = y⟨1⟩, y⟨2⟩, …. y⟨T^y⟩).

$$\hat{Y} = \arg\max_{Y} P(Y|X)$$

If we elect Beam Width = B = 3, within the initiative , the model evaluates the probability of the primary word, given only input X i.e. P(y‹1›|X). During this step, the input Hindi sentence is run through the encoder LSTM and therefore the initiative of the decoder LSTM are going to be a softmax output over all the chances within the English vocabulary. Now for the primary word, of all the likely translations, top three are picked. The algorithm stores these three choices for subsequent steps.

In this step, for every of the three options picked, subsequent choice is estimated i.e.

P(y‹2›|X , y‹1›)

The network hard-wires the primary word y‹1›. After this step, we've the contingent probability calculated as:

$$P(\hat{y}_{\langle 1\rangle}, \hat{y}_{\langle 2\rangle}|X) = P(\hat{y}_{\langle 1\rangle}|X) \cdot P(\hat{y}_{\langle 2\rangle}|X, \hat{y}_{\langle 1\rangle})$$

The complete code for implementing beam search are often found here.

Eg- how predictions displayed using beam search:

Source: कालबैक फोन

Target: callback phone

VI. RESULTS AND DISCUSSION

The initial requirement of setting up a translation is the availability of corpus for target languages and source. A resourceful language is not Hindi.

Neural machine translation is implemented by core algorithm from tutorials from Peter Neubig. There are the platforms- Tensor Flow and Theano.

A. BLEU Score

We evaluated our system using BLEU score. In each the score of the translation score are different and for each different configuration shows the score.

B. Discussions

The results are obtained from neural machine translation from English to Hindi machine translation is comparable with statistical or phrase based machine translation.

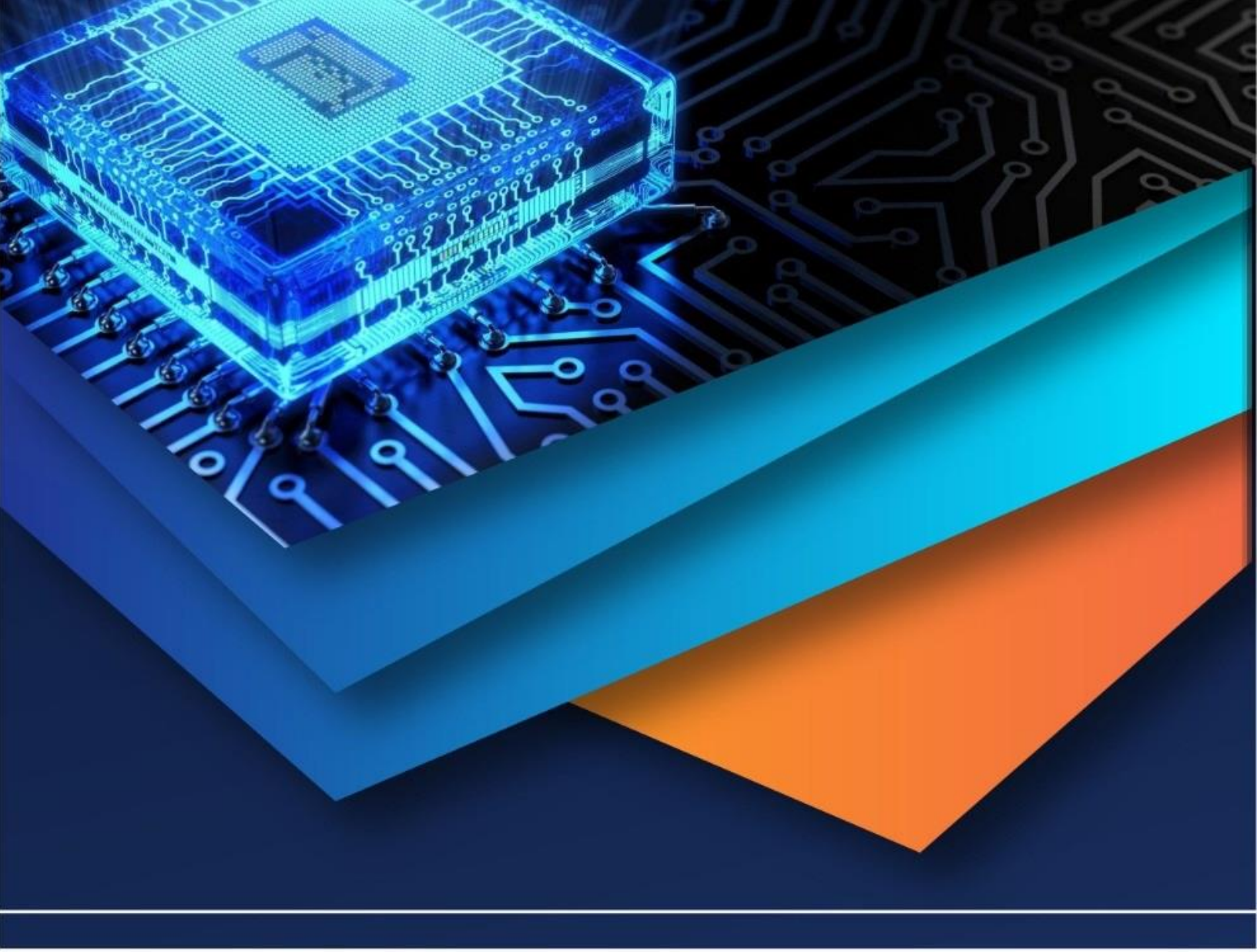
In our work, we have inspired task of translation from human abilities to translate the language.

VII. CONCLUSION

Statistical phrase machine translation system has been facing the problem of requirement of large data sets for a long time and accuracy. And in this work we investigated the possibility of using a shallow RNN and LSTM based neural machine translator for solving the issue of machine translation.

REFRENCES

[1] https://books.google.com/books?hl=en&lr=&id=kKYgAwAAQBAJ&oi=fnd&pg=PR11&dq=machine+translation&ots=k6Eo_X7-3q&sig=S0lENmoKJRulDBcZybIyDbDyGIA
[2] https://pdfs.semanticscholar.org/b6fb/e874f9c8aa775e6a9722a4d0c613385712b0.pdf
[3] https://medium.com/analytics-vidhya/neural-machine-translation-for-hindi-english-sequence-to-sequence-learning-1298655e334a
[4] https://en.wikipedia.org/wiki/Neural_machine_translation
[5] https://builtin.com/artificial-intelligence
[6] http://dynamicpublisher.org/gallery/ijsrr-d121.pdf
[7] https://ieeexplore.ieee.org/abstract/document/8464781/
[8] http://www.columbia.edu/itc/hs/medinfo/g6080/misc/articles/friedman.pdf
[9] S. Saini, U. Sehgal, and V. Sahula.Relative clause based text simplification for improved english to hindi translation. In 2015International Conference on Advances in Computing, Communications and Informatics (ICACCI), pages 1479–1484, Aug 2015.
[10] S. Chand. Empirical survey of machine translation tools. In 2016Second International Conference on Research in Computational Intelli-gence and Communication Networks (ICRCICN), pages 181–185, Sept2016.
[11] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. CoRR, abs/1409.3215, 2014.
[12] Sepp Hochreiter and J¨urgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
[13] Graham Neubig. Neural machine translation and sequence-to-sequencemodels: A tutorial. arXiv preprint arXiv:1703.01619, 2017.
[14] LR Medsker and LC Jain. Recurrent neural networks. Design and Applications, 5, 2001.
[15] Sepp Hochreiter and J¨urgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997
[16] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neuralnetworks. IEEE Transactions on Signal Processing, 45(11):2673–2681,1997.
[17] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficultyof training recurrent neural networks. In International Conference onMachine Learning, pages 1310–1318, 2013.
[18] Institute for language, cognition and computation,university of edinburgh, indic multi-parallel corpus,http://homepages.inf.ed.ac.uk/miles/babel.html. Technical report



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY